

# UC San Diego

## UC San Diego Previously Published Works

### Title

Pgltools: a genomic arithmetic tool suite for manipulation of Hi-C peak and other chromatin interaction data.

### Permalink

<https://escholarship.org/uc/item/5dp72331>

### Journal

BMC bioinformatics, 18(1)

### ISSN

1471-2105

### Authors

Greenwald, William W  
Li, He  
Smith, Erin N  
et al.

### Publication Date

2017-04-01

### DOI

10.1186/s12859-017-1621-0

Peer reviewed

SOFTWARE

Open Access



# Pgltools: a genomic arithmetic tool suite for manipulation of Hi-C peak and other chromatin interaction data

William W. Greenwald<sup>1</sup>, He Li<sup>2,3</sup>, Erin N. Smith<sup>3</sup>, Paola Benaglio<sup>3</sup>, Naoki Nariai<sup>2,3</sup> and Kelly A. Frazer<sup>2,3\*</sup>

## Abstract

**Background:** Genomic interaction studies use next-generation sequencing (NGS) to examine the interactions between two loci on the genome, with subsequent bioinformatics analyses typically including annotation, intersection, and merging of data from multiple experiments. While many file types and analysis tools exist for storing and manipulating single locus NGS data, there is currently no file standard or analysis tool suite for manipulating and storing paired-genomic-loci: the data type resulting from “genomic interaction” studies. As genomic interaction sequencing data are becoming prevalent, a standard file format and tools for working with these data conveniently and efficiently are needed.

**Results:** This article details a file standard and novel software tool suite for working with paired-genomic-loci data. We present the **paired-genomic-loci (PGL)** file standard for genomic-interactions data, and the accompanying analysis tool suite “pgltools”: a cross platform, pypy compatible python package available both as an easy-to-use UNIX package, and as a python module, for integration into pipelines of paired-genomic-loci analyses.

**Conclusions:** Pgltools is a freely available, open source tool suite for manipulating paired-genomic-loci data. Source code, an in-depth manual, and a tutorial are available publicly at [www.github.com/billgreenwald/pgltools](http://www.github.com/billgreenwald/pgltools), and a python module of the operations can be installed from PyPI via the PyGLtools module.

**Keywords:** Hi-CChIA-PET, Chromatin conformation capture, Peak, Paired-genomic-loci, Tool suite, Bedtools, Genomic arithmetic

## Background

Numerous experimental methodologies have been developed in the past decade to study 3D configurations of the human genome, including Hi-C and ChIA-PET [1, 2]. These “genomic interaction” data have provided key insights into the regulation of gene expression, and suggest that chromatin interactions are driven by discrete, yet spatially-associated, epigenetic features [3, 4]. File standards and tool suites have become essential to conduct efficient bioinformatics analyses; for example, single locus information can be encoded in the BED file format and manipulated using bedtools, enabling a wide variety of bioinformatics inquiries [5]. However, it is

currently challenging to fully interpret the biological impact of genomic interactions as tools do not yet exist to quickly and iteratively interrogate the extent to which both regions of paired loci are conserved across genomic datasets from diverse cell-types and contexts. While paired-genomic-loci data generated from these methodologies are widely available, the bioinformatics field has not yet developed either a file standard or analysis tools for their efficient manipulation.

There are currently several file formats for paired-genomic-loci data, however, none of these file formats were designed to enable efficient annotation and data manipulation. Existing file formats include those that encode read count information such as the matrix and the triplet sparse matrix formats [6], and others that encode the locations of paired segments and specialized metadata for particular pipelines, such as the HiFive ChromatinInteraction format [7]. Although the matrix and triplet sparse matrix formats effectively communicate coverage depth across

\* Correspondence: [kafrazer@ucsd.edu](mailto:kafrazer@ucsd.edu)

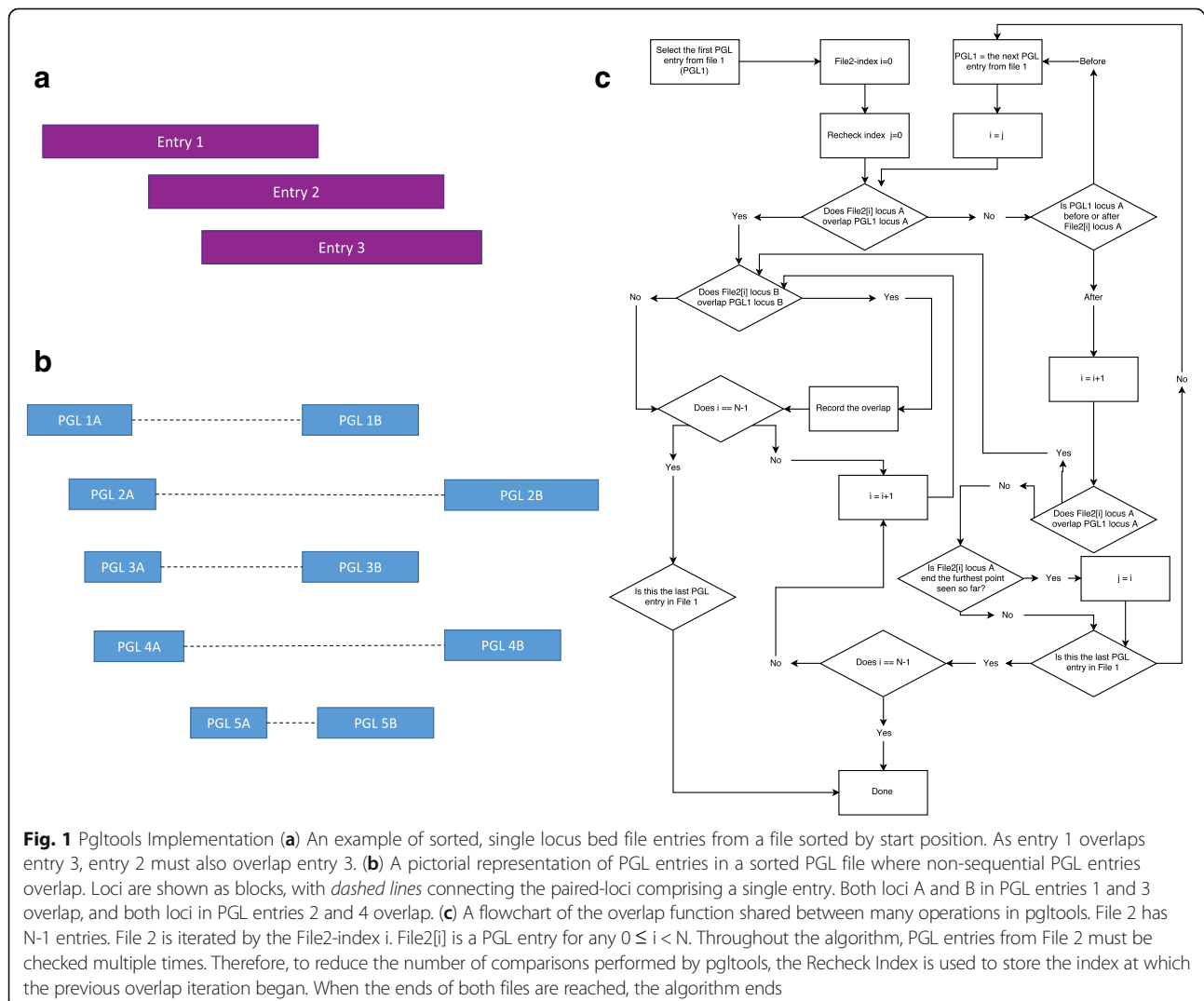
<sup>2</sup>Institute for Genomic Medicine, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA

<sup>3</sup>Department of Pediatrics and Rady Children's Hospital, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA

Full list of author information is available at the end of the article



Here we describe a new file standard for **paired-genomic-loci** data, the PGL format, and an analysis tool suite, **pgltools**, for genomic interaction data storage and manipulation. The PGL format supports genomic interaction data, allows for appropriate metadata, and enables efficient data manipulation. Pgltools performs genomic



arithmetic on PGL files such as comparing, merging, and intersecting two sets of paired-genomic-loci, as well as integrates BED files with PGL files. Finally, we provide functions to convert other genomic interaction file formats to PGL files, and convert PGL files to multiple different visualization formats. This analysis tool suite will allow for iterative bioinformatics analyses and visualization of genomic interaction data, facilitating discovery and collaboration within the genomic interaction field.

## Implementation

Our goal was to create a file standard that can summarize the output from mapping and peak calling algorithms for chromatin interaction data derived from experiments, such as Hi-C or ChIA-PET, that is easily interpretable, shareable, and can be combined with current genomic annotation formats, such as the BED format. We first established a paired-genomic-loci file standard—the “PGL” file type—which represents each paired-genomic-loci as a single PGL entry in a human readable text file, with space in each entry for annotations, and then implemented an analysis tool suite for working with these files. Within “genomic interaction” data, the interactions between two loci (locus A and locus B) are captured—this “paired” information is preserved through the PGL file standard. PGL files require six columns in the following order: locus A chromosome, locus A start position, locus A end position, locus B chromosome, locus B start position, and locus B end position. Beyond the six columns, any user-defined annotations, such as interaction *p*-value or locus chromatin state, can be written. These annotations can be manipulated and utilized by the operations in PGLtools to gain insight into the relationship between multiple paired-genomic-loci. As annotations are unique to a file, headers can be given in files by preceding a line with “#.” Furthermore, PGL files are required to have each PGL entry written such that locus A comes before locus B based on chromosome number alphabetically (ex. chr1, chr10, chr15, chr22, chr7, chrX, chrY) and chromosome position numerically. This relationship, when combined with file sorting on each column sequentially, gives pgltools the ability to quickly merge and intersect PGL entries from PGL files. Operations for sorting PGL files, converting files to PGL files, and formatting PGL files for visualization with established programs, are also included in pgltools.

Most pgltools operations utilize the same core overlap function to test for overlapping paired-genomic-loci within or between file(s). For single locus entries, such as those in sorted BED files, overlapping entries must be sequential: if entries 1 and 3 overlap, entry 2 must overlap both entries 1 and 3 (Fig. 1a). This property allows bedtools to limit the number of features that must be compared for overlap, thus expediting analyses [5]. However, in sorted PGL files, while locus A from multiple

sequential entries can overlap, locus B may not overlap (Fig. 1b). The pgltools overlap function allows for this and quickly and efficiently finds consecutive and non-consecutive entries where both locus A and locus B are overlapping. It begins by comparing the first PGLs in both files, recording if an overlap occurred in both loci, and then advances to the next PGL in File 2. These comparisons continue until the PGL from File 2 does not overlap locus A from the PGL in File 1, at which point the algorithm begins comparing the next PGL from File 1 to the first possible overlapping PGL from File 2. This repeats until the ends of both files are reached. An in-depth flow chart of the overlap operation’s control flow,

**Table 1** Summary of operations provided in pgltools

Method	Description
intersect	Find overlapping paired-genomic-loci from two PGL files
merge	Merge nearby paired-genomic-loci within a single file and produce a column containing summary statistics requested through passed parameters (-c and -o)
subtract	Find parts of paired-genomic-loci from a PGL file that do not overlap another PGL file
window	Filter a PGL file to a particular genomic region
samToPgl	Converts a sam file to a PGL file
coverage	Find the coverage of a PGL file on another PGL file; usually used to find the coverage of reads from a PGL file derived from a sam file on a set of PGLs. The paired-genomic-loci from file 2 only need to overlap the paired-genomic-loci from file 1.
closest	Find the closest paired-genomic-loci from a PGL file for each paired-genomic-loci in another PGL file
expand	Expand both loci by a given size
intersect1D	Find the paired-genomic-loci that overlap regions from a bed file
closest1D	Find the closest paired-genomic-loci to a set of regions from a bed file
subtract1D	Find the parts of paired-genomic-loci that do not overlap regions from a bed file
sort	Sorts a PGL file for use with other PGLtools operations
formatbedpe	Convert a bedpe-like file to a PGL file
formatTripSparse	Convert a triplet sparse matrix file set to a PGL file
converT	Formats the PGL file for use with the GenomicInteractions R package
browser	Format a PGL file to be viewed in the UCSC Genome Browser
juicebox	Format a PGL file to be viewed in juicebox
condense	Convert a PGL file to a BED file with two entries for each PGL entry.
findLoops	Convert a PGL file to a BED file with an entry containing the region from the start of anchor A to the stop of anchor B for intra-chromosomal PGLs, and an entry for each anchor for inter-chromosomal PGLs.

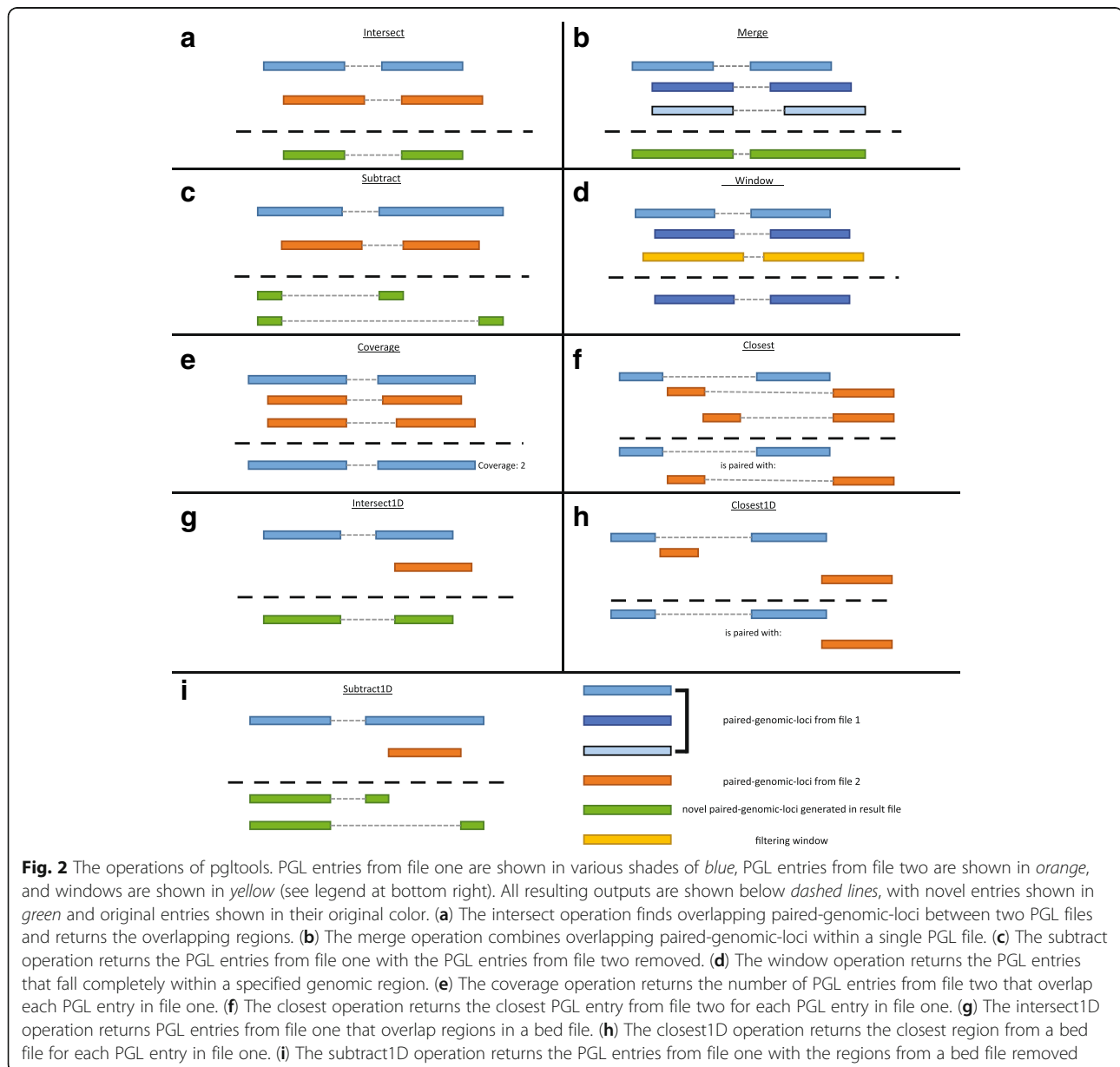
as well as how the first possible overlapping PGL from File 2 is determined, is shown in Fig. 1c.

Pgltools is implemented in Python 2.7, and all operations have been tested with the pypy python compiler. As such, the UNIX package version of pgltools can be run either with CPython or pypy; the included UNIX wrapper will run pgltools through pypy if installed, or CPython if pypy is not installed. Utilizing pypy reduces memory consumption by approximately 25%, and decreases run times 5–7 fold. The pgltools suite can read from UNIX standard in, useful for stringing multiple pgltools commands together without needing to save the intermediate files, and writes to UNIX standard out, allowing it to be utilized in complex pipelines to speed

up analysis of genomic interaction data. Pgltools is also available as a python module, PyGLtools, for use within pythonic pipelines, and can be installed from PyPI. As pgltools is written in Python 2.7, it is easily portable to any platform and poised for collaboration with the community.

## Results and Discussion

Table 1 includes a full list of pgltools operations and their default behavior. Visualizations of these operations are provided in Fig. 2. The pgltools *intersect* operation can be used to identify either the overlap, union, or uniqueness of PGL entries between two PGL files, while preserving or combining annotations during these analyses; for example,



the number of overlapping bases at each locus from each PGL entry from two PGL files can be determined. The *pgltools merge* operation can be utilized to merge overlapping PGL entries, or PGL entries within a specified distance within a single PGL file. Summary statistics, such as the number of merged entries, can be obtained through command line arguments to the *merge* operation. To determine differential PGL entries between two PGL files, the *subtract* operation has been included to remove the parts of PGL entries present in one PGL file from those present in another. Once a set of PGL entries has been determined, it is common to filter these entries to a desired genomic region—the *window* operation can be used to filter based on either or both end(s) of the PGL entries in a PGL file. To interrogate questions regarding differential coverage depth of genomic interactions, such as genetic association with interaction intensity, we provide the *sam-Topgl* operation, which when utilized with the *coverage* operation, will find the number of reads from a sam file that overlap each PGL entry in a PGL file (though the operation is generalizable for any two PGL files). The *closest* operation is provided for finding the closest PGL entries between two PGL files. The *expand* operation can expand both loci by a given value. In addition, as single locus genomic metadata is often analyzed together with interaction data, such as presence of a coding region, epigenetic annotation, or motif locations, we provide the *intersect1D*, *closest1D*, and *subtract1D* operations for analysis on traditional BED files and PGL files. Finally, we include helper operations both for converting files to the PGL format, including *formatbedpe* to convert a bedpe file and *formatTripSparse* to convert triple sparse matrix files, and for converting from the PGL format to packages for visualization or further analysis, such as the *conveRt* operation to convert to a file readable by the GenomicInteractions R package [10], *browser* for visualizing with the UCSC Genome Browser [11], *juiceBox* for visualizing with JuiceBox [3, 12], and *condense* and *findLoops* to create a BED file of either the discrete loci or interior regions of each PGL.

By combining the operations within *pgltools*, one can quickly and easily interrogate biological functionality in the context of chromatin interaction data. For example, by combining the *intersect1D* and *merge* operations, it is possible to determine the different chromatin annotations for each locus of each PGL entry (which could then be further filtered to determine 3D interactions between chromatin states of interest, e.g. promoter-enhancer). Additionally, *pgltools* can be used to find overlaps between chromatin interactions and other types of paired data. For example, one could create a PGL file from a list of expression Quantitative Trait Loci (eQTLs) and their corresponding target genes (eGenes), and utilize the *intersect* operation to determine if any pairs

of eQTL and eGenes fall within a chromatin interaction. Example pipelines for these scenarios can be found on the *pgltools* github.

## Conclusions

*Pgltools* is an open source software analysis tool suite for interacting with the PGL file standard for paired-genomic-loci. *Pgltools* can read from and writes to UNIX standard in and standard out, and can be run quickly in both CPython and pypy. A python module version, *PyGLtools*, is available for use within pythonic pipelines. The cross-platform nature of python poises *pgltools* for community contribution, and makes it easy to install and utilize.

## Abbreviations

eQTLs: Expression quantitative trait loci; NGS: Next-generation sequencing; PGL: Paired-genomic-loci

## Acknowledgements

We would like to acknowledge Anthony D. Schmitt for aiding in generating pre-published data for software development and unit testing.

## Funding

This work was supported in part by a California Institute for Regenerative Medicine grant [GC1R-06673 to KAF]; and National Institutes of Health grants [HG008118 to KAF, HL107442 to KAF, DK105541 to KAF, DK112155 to KAF and EY021237 to KAF]. These funding agencies played no role in the design or conclusions of this study.

## Availability of data and materials

Test datasets and software generated and analyzed during the current study are available in the *pgltools* repository at <https://www.github.com/billgreenwald/pgltools>, and a python module implementation (*PyGLtools*) can be downloaded from PyPI.

## Authors' contributions

WWG and HL conceived of the project. WWG wrote the code. HL and NN unit tested and wrote documentation. PB generated the data utilized for testing. KAF directed the project. WWG, HL, NN, ENS and KF wrote the manuscript. All authors read and approved the final manuscript.

## Competing interests

The authors declare that they have no competing interests.

## Consent for publication

Not Applicable.

## Ethics approval and consent to participate

Not Applicable.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Author details

<sup>1</sup>Bioinformatics and Systems Biology, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. <sup>2</sup>Institute for Genomic Medicine, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA. <sup>3</sup>Department of Pediatrics and Rady Children's Hospital, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093, USA.

Received: 16 December 2016 Accepted: 31 March 2017

Published online: 07 April 2017

## References

- Li G, Fullwood MJ, Xu H, Mulawadi FH, Velkov S, Vega V, Ariyaratne PN, Bin MY, Ooi H, Tennakoon C, Wei C. ChIA-PET tool for comprehensive

- chromatin interaction analysis with paired-end tag sequencing. *Genome Biol.* 2010;11:R22.
2. Lieberman-Aiden E, Berkum NL Van, Williams L, Imakaev M, Ragoczy T, Telling A, Amit I, Lajoie BR, Sabo PJ, Dorschner MO, Sandstrom R, Bernstein B, Bender MA, Groudine M, Gnirke A, Stamatoyannopoulos J, Mirny LA. Comprehensive Mapping of Long-Range Interactions Reveals Folding Principles of the Human Genome. *Science.* 2009;326:289–94.
  3. Rao SSP, Huntley MH, Durand NC, Stamenova EK. A 3D Map of the Human Genome at Kilobase Resolution Reveals Principles of Chromatin Looping. *Cell.* 2014;159:1665–80.
  4. Dixon JR, Jung I, Selvaraj S, Shen Y, Antosiewicz-Bourget JE, Lee AY, Ye Z, Kim A, Rajagopal N, Xie W, Diao Y, Liang J, Zhao H, Lobanenkov WV, Ecker JR, Thomson JA, Ren B. Chromatin architecture reorganization during stem cell differentiation. *Nature.* 2015;518:331–6.
  5. Quinlan AR, Hall IM. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics.* 2010;26:841–2.
  6. Servant N, Varoquaux N, Lajoie BR, Viara E, Chen C, Vert J, Heard E, Dekker J, Barillot E. HiC-Pro: an optimized and flexible pipeline for Hi-C data processing. *Genome Biol.* 2015;16:259.
  7. Sauria MEG, Phillips-cremins JE, Corces VG, Taylor J. HiFive: a tool suite for easy and efficient HiC and 5C data analysis. *Genome Biol.* 2015;16:237.
  8. Durand NC, Shamim MS, Machol I, Rao SSP, Huntley MH, Lander ES, Aiden EL, Durand NC, Shamim MS, Machol I, Rao SSP, Huntley MH. Juicer Provides a One-Click System for Analyzing Tool Juicer Provides a One-Click System for Analyzing Loop-Resolution Hi-C Experiments. *Cell Syst.* 2016;3:95–8.
  9. Ay F, Bailey TL, Noble WS: Statistical confidence estimation for Hi-C data reveals regulatory chromatin contacts. *Genome Res* 2014;1–24. doi:10.1101/gr.160374.113
  10. Harmston N, Ing-Simmons E, Perry M, Barešić A, Lenhard B. GenomicInteractions: An R/Bioconductor package for manipulating and investigating chromatin interaction data. *BMC Genomics.* 2015;16:963.
  11. Kent WJ, Sugnet CW, Furey TS, Roskin KM. The Human Genome Browser at UCSC W. *Genome Res.* 2002;12:996–1006.
  12. Durand NC, Robinson JT, Shamim MS, Machol I, Mesirov JP, Lander ES, Durand NC, Robinson JT, Shamim MS, Machol I, Mesirov JP. Juicebox Provides a Visualization System for Hi-C Contact Maps with Unlimited Zoom Tool Juicebox Provides a Visualization System for Hi-C Contact Maps with Unlimited Zoom. *Cell Syst.* 2016;3:99–101.

Submit your next manuscript to BioMed Central and we will help you at every step:

- We accept pre-submission inquiries
- Our selector tool helps you to find the most relevant journal
- We provide round the clock customer support
- Convenient online submission
- Thorough peer review
- Inclusion in PubMed and all major indexing services
- Maximum visibility for your research

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)

